



```

RRRRRRRR  MM   MM   SSSSSSSS  000000  CCCCCCCC  RRRRRRRR  EEEEEEEE  AAAAAAA  TTTTTTTT
RRRRRRRR  MM   MM   SSSSSSSS  000000  CCCCCCCC  RRRRRRRR  EEEEEEEE  AAAAAAA  TTTTTTTT
RR   RR  MMMM  MMMM  SS   00   00  CC   RR   RR  EE   AA   AA  TT
RR   RR  MMMM  MMMM  SS   00   00  CC   RR   RR  EE   AA   AA  TT
RR   RR  MM   MM   MM  SS   00   0000  CC   RR   RR  EE   AA   AA  TT
RR   RR  MM   MM   MM  SS   00   0000  CC   RR   RR  EE   AA   AA  TT
RRRRRRRR  MM   MM   SSSSSS  00   00  00  CC   RRRRRRRR  EEEEEEEE  AA   AA  TT
RRRRRRRR  MM   MM   SSSSSS  00   00  00  CC   RRRRRRRR  EEEEEEEE  AA   AA  TT
RR   RR  MM   MM   SS  0000  00  CC   RR   RR  EE   AAAAAAAA  TT
RR   RR  MM   MM   SS  0000  00  CC   RR   RR  EE   AAAAAAAA  TT
RR   RR  MM   MM   SS  00   00  CC   RR   RR  EE   AA   AA  TT
RR   RR  MM   MM   SS  00   00  CC   RR   RR  EE   AA   AA  TT
RR   RR  MM   MM   SSSSSSSS  000000  CCCCCCCC  RR   RR  EEEEEEEE  AA   AA  TT
RR   RR  MM   MM   SSSSSSSS  000000  CCCCCCCC  RR   RR  EEEEEEEE  AA   AA  TT

```

(2) 105 DECLARATIONS  
(3) 133 RMS\$CREATE - SCREATE ROUTINE

0000 1 \$BEGIN RMSOCREAT,000,RMSRMS,<DISPATCH FOR CREATE OPERATION>  
0000 2  
0000 3  
0000 4 \*\*\*\*\*  
0000 5 \*  
0000 6 \* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
0000 7 \* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
0000 8 \* ALL RIGHTS RESERVED.  
0000 9 \*  
0000 10 \* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
0000 11 \* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
0000 12 \* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
0000 13 \* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
0000 14 \* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
0000 15 \* TRANSFERRED.  
0000 16 \*  
0000 17 \* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
0000 18 \* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
0000 19 \* CORPORATION.  
0000 20 \*  
0000 21 \* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
0000 22 \* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
0000 23 \*  
0000 24 \*  
0000 25 \*\*\*\*\*  
0000 26  
0000 27 ++  
0000 28 Facility: rms32  
0000 29  
0000 30 Abstract:  
0000 31 this module is the highest level control routine  
0000 32 to perform the \$create function.  
0000 33  
0000 34 Environment:  
0000 35 star processor running starlet exec.  
0000 36  
0000 37 Author: L F LAVERDURE, Creation Date: 8-MAY-1977  
0000 38  
0000 39 Modified By:  
0000 40  
0000 41 V03-008 RAS0286 Ron Schaefer 3-Apr-1984  
0000 42 Revise the search!st continuation rules/errors.  
0000 43  
0000 44 V03-007 RAS0244 Ron Schaefer 23-Jan-1984  
0000 45 Add \$DEVDEF macro.  
0000 46  
0000 47 V03-006 RAS0198 Ron Schaefer 6-Oct-1983  
0000 48 Merge \$OPEN and SCREATE searchlist looping logic into  
0000 49 a single routine. Update error checking logic to only  
0000 50 loop for certain errors, not for all.  
0000 51  
0000 52 V03-005 RAS0197 Ron Schaefer 5-Oct-1983  
0000 53 Add support for searchlists with create-if. The  
0000 54 method is to try straight-forward \$OPEN on ALL elements  
0000 55 of the search list. If no file is found anywhere, then  
0000 56 do a create-if ACP operation on the first location.  
0000 57

0000	58	:	V03-004	JWH0215	Jeffrey W. Horn	13-Apr-1983
0000	59	:		Fix broken branch.		
0000	60	:				
0000	61	:	V03-003	LJA0021	Laurie Anderson	3-Sep-1982
0000	62	:		If this is a Restart create operation, then just go and		
0000	63	:		open the file, using OPEN.		
0000	64	:				
0000	65	:	V03-002	KBT0175	Keith B. Thompson	23-Aug-1982
0000	66	:		Reorganize psects and make entry point single '\$'		
0000	67	:				
0000	68	:	V03-001	RAS0084	Ron Schaefer	2-Apr-1982
0000	69	:		Force stream format files to have RAT non null.		
0000	70	:				
0000	71	:	V02-018	CDS0001	C Saether	5-Feb-1982
0000	72	:		Store GBC in ifab from fab.		
0000	73	:				
0000	74	:	V02-017	JAK0068	J A Krycka	29-DEC-1981
0000	75	:		Preserve alternate success code on network exit path.		
0000	76	:				
0000	77	:	V02-016	MCN0005	Maria del C. Nasr	10-Feb-1981
0000	78	:		Make sure magtape is mounted before creating file.		
0000	79	:				
0000	80	:	V02-015	JAK0048	J A KRYCKA	25-SEP-1980
0000	81	:		Move network specific code from RMOCRECOM to this module and		
0000	82	:		add support for the create-if (CIF) option.		
0000	83	:				
0000	84	:	V014	REFORMAT	D M WALP	25-JUL-1980
0000	85	:				
0000	86	:	V013	RAS0003	R A SCHAEFER	27-NOV-1979
0000	87	:		allow any record format for bio access.		
0000	88	:				
0000	89	:	V012	TMH0001	TIM HALVORSEN	13-OCT-1979
0000	90	:		fix wildcard check to use the wildcard summary bit		
0000	91	:		rather than checking explicitly for wc name,type,ver.		
0000	92	:				
0000	93	:	V011	RAN0002	R A NEWELL	1-SEP-1978 12:00
0000	94	:		rms32 isam modification. redefinition of entry points to		
0000	95	:		resolve out of range branches.		
0000	96	:				
0000	97	:	V010	JAK0001	J A KRYCKA	24-AUG-1978 07:53
0000	98	:		miscellaneous clean-up prior to decnet v1.0 code freeze.		
0000	99	:		skip organization dependent code on network operations.		
0000	100	:				
0000	101	:	--			
0000	102	:				
0000	103	:				

```
0000 105      .SBTTL DECLARATIONS
0000 106
0000 107      :
0000 108      ; Include Files:
0000 109      :
0000 110
0000 111      :
0000 112      ; Macros:
0000 113      :
0000 114
0000 115      $DEVDEF
0000 116      $IFBDEF
0000 117      $FIBDEF
0000 118      $FABDEF
0000 119      $FWADEF
0000 120      $RMSDEF
0000 121
0000 122      :
0000 123      ; Equated Symbols:
0000 124      :
0000 125
00000020 0000 126      FOP=    FAB$L_FOP*8      ; bit offset to file options longword
0000 127
0000 128      :
0000 129      ; Own Storage:
0000 130      :
0000 131
```

```
0000 133 .SBTTL RMS$CREATE - $CREATE ROUTINE
0000 134
0000 135 ++
0000 136 RMS$CREATE
0000 137
0000 138 RMS$CREATE -
0000 139
0000 140      this routine performs the highest level $create processing.
0000 141      its functions include:
0000 142
0000 143      1. common setup and checks
0000 144      2. dispatch to organization-dependent code
0000 145      which calls common create routines
0000 146
0000 147
0000 148 Calling sequence:
0000 149
0000 150      entered from exec as a result of user's calling sys$create
0000 151      (e.g., by using the $create macro).
0000 152
0000 153 Input Parameters:
0000 154
0000 155      ap      user's argument list addr
0000 156
0000 157 Implicit Inputs:
0000 158
0000 159      the contents of the fab and possible related user interface
0000 160      blocks.
0000 161
0000 162 Output Parameters:
0000 163
0000 164      r0      status code
0000 165      r1      destroyed
0000 166
0000 167 Implicit Outputs:
0000 168
0000 169      the various fields of the fab are filled in to reflect
0000 170      the status of the created open file. (see rms functional spec for
0000 171      a complete list.)
0000 172      an ifab is initialized to reflect the created file.
0000 173
0000 174      a completion ast is queued if so specified by the user.
0000 175
0000 176 Completion Codes:
0000 177
0000 178      standard rms (see functional spec for list).
0000 179
0000 180 Side Effects:
0000 181
0000 182      none
0000 183
0000 184      --
0000 185
```

03 69 3B FFF7' 30 0006 0000 187 SENTRY RMSS\$CREATE  
0009 0000 188 STSTPT CREATE  
0009 0009 189 BSBW RM\$FSETI ; do common setup  
0010 000D 191 BBC #IFBSV\_RESTART,(R9),3\$ ; See if this is a restart operation.  
0010 0010 192 BRW RM\$OPEN\_ALT ; Yes, just make it an open call.  
0010 0010 193  
0010 0010 194 :  
0010 0010 195 :NOTE: DOES NOT RETURN ON ERROR  
0010 0010 196  
0010 0010 197 : an ifab has been set up  
0010 0010 198 : perform various checks  
0010 0010 199 :  
0010 0010 200  
16 A8 01 89 0014 0010 201 3\$: SSB #IFBSV\_CREATE,(R9) ; remember we're doing create  
22 A9 02 05 ED 001A 0014 202 BISB3 #FABSM\_PUT,FAB\$B\_FAC(R8),-  
03 04 12 0020 0018 203 IFBSB\_FAC(R9) ; imply put access  
22 A9 20 8A 0022 001A 204 ASSUME FAB\$V\_BRO EQ FAB\$V\_BIO+1  
0026 0026 205 CMPZV #FAB\$V\_BIO,#2,IFBSB\_FAC(R9),-  
0026 0026 206 #3 ; both bro & bio set?  
0026 0026 207 BNEQ 5\$ ; branch if not  
0026 0026 208 BICB2 #FABSM\_BIO,IFBSB\_FAC(R9); clear bio (implied  
0026 0026 209 : by bro without restrictions)  
0026 0026 210 5\$:  
0026 0026 211  
0026 0026 212 :  
0026 0026 213 : validate rfm  
0026 0026 214 :  
0026 0026 215 :  
1F A8 50 A9 90 0026 216 ASSUME IFBSV\_RFM EQ 0  
0026 0026 217 ASSUME IFBS\$-RFM EQ 4  
0026 0026 218 MOVB FAB\$B\_RFM(R8),-  
0026 0026 219 IFBSB\_RFMORG(R9) ; save rfm in rfmorg  
0028 0028 220  
0028 0028 221 : check for rfm in supported range  
0028 0028 222 :  
0028 0028 223 :  
0028 0028 224 :  
06 09 22 A9 05 E0 002B 225 BBS #IFBSV\_BIO,-  
1F A8 03 00B3 91 0030 002D 226 IFBSB\_FAC(R9),10\$ ; don't check for bio  
03 31 0034 0030 227 CMPB FAB\$B\_RFM(R8),#FABSC\_MAXRFM  
0036 0034 228 BLEQU 10\$  
0039 0036 229 BRW ERRRFM  
0039 0039 230  
0039 0039 231 : set mrs, gbc, lrl, rat, and fsz from fab  
0039 0039 232 :  
0039 0039 233 :  
0039 0039 234 :  
01 36 A8 60 A9 80 0039 235 10\$: MOVW FAB\$W\_MRS(R8),-  
1F A8 91 003C 003C 236 IFBSW\_MRS(R9) ; save mrs  
0A 12 0042 003E 237 CMPB FAB\$B\_RFM(R8),#FABSC\_FIX ; fixed length rfm?  
36 A8 80 0044 238 BNEQ 20\$ ; branch if not  
52 A9 0047 239 MOVW FAB\$W\_MRS(R8),-  
03 14 0049 240 IFBSW\_LRL(R9) ; save fixed length  
00A6 31 004B 241 BGTR 20\$  
48 A8 80 004E 242 BRW ERRMRS ; branch if not > 0  
004E 243 20\$: MOVW FAB\$W\_GBC(R8),- ; store gbc

```

64 A9 0051 244
03 0053 245
00A0 31 0055 246 BGEQ IFBSW_GBC(R9)
0058 247 21$: BRW 21$ ; ok if non-negative
0058 248 ERRGBC ; else it's an error
0058 249 ; force stream format files to have RAT non-null.
0058 250
0058 251 ASSUME FABSC_STM LT FABSC_STMLF
0058 252 ASSUME FABSC_STM LT FABSC_STMCR
0058 253
1F AB 91 0058 254 CMPB FABSB_RFM(R8),- ; stream format?
04 005B 255 #FABSC_STM
0A 1F 005C 256 BLSSU 22$ ; nope
07 93 005E 257 BITB #<FABSM_CR!FABSM_FTN!FABSM_PRN>,-
1E A8 0060 258 FABSB_RAT(R8) ; carriage control already set?
04 12 0062 259 BNEQ 22$ ; ok
1E A8 02 88 0064 260 BISB2 #FABSM_CR,FABSB_RAT(R8) ; force RAT=CR
1E A8 90 0068 261 22$: MOVB FABSB_RAT(R8),- ; save rat
51 A9 0068 262 IFBSB_RAT(R9)
006D 263
006D 264 ; check that only one of 'prn', 'cr' and 'ftn' rat bits are on
006D 265
006D 266 ; zero-extend the field
006D 267
51 1E A8 51 D4 006D 268 CLRL R1
06 51 F8 8F 8B 006F 269 BICB3 #255\<FABSM_CR!FABSM_FTN!FABSM_PRN>,FABSB_RAT(R8),R1
02 E1 0075 270 BBC #FABSV_PRN,R1,23$ ; branch if not prn
03 1F A8 91 0079 271 CMPB FABSB_RFM(R8),#FABSC_VFC ; rfm = vfc?
69 12 007D 272 BNEQ ERRRAT ; branch if not
007F 273 ASSUME FABSV_FTN EQ 0
007F 274 ASSUME FABSV_CR EQ FABSV_FTN+1
007F 275 ASSUME FABSV_PRN EQ FABSV_CR+1
64 17'8F 51 E1 007F 276 23$: BBC R1,I^#^B00010111,ERRRAT ; branch if yes
0084 277
0084 278 ; if vfc record format, check for 0 fixed header size and if
0084 279 found make it 2 bytes
0084 280
0084 281
0084 282
03 1F A8 91 0084 283 CMPB FABSB_RFM(R8),#FABSC_VFC
16 12 0088 284 BNEQ 30$ ; omit check if not vfc
3F A8 90 008A 285 MOVB FABSB_FSZ(R8),-
5F A9 008D 286 BNEQ 29$ ; save fsz value
04 12 008F 287 MOVBL FABSB_FSZ(R9)
29$: BNEQ 29$ ; branch if value specified
5F A9 02 90 0091 288 MOVBL #2,IFBSB_FSZ(R9) ; else set default value
02 06 51 A9 0095 289 29$: BBC #FABSV_PRN,-
5F A9 91 009A 290 BNEQ IFBSB_RAT(R9),30$ ; branch if not prn
50 1F 009E 291 CMPB IFBSB_FSZ(R9),#2 ; fsz > 1?
00A0 292 BLSSU BERRFSZ ; branch if not
FF5D' 30 00A0 294 BSBW RMSPRFLNMAINT ; process file name
65 50 E9 00A3 295 BLBC R0,EXIT ; exit on error
6A 18 E0 00A6 296 BBS #FWASV_WILDCARD,(R10),-
56 00A9 297 BBS ERRSYN ; no wildcards allowed
6A 38 E1 00AA 298 CIF: BBC #FWASV_SLPRESENT,(R10),- ; no problem if no searchlist
04 00AD 299 SETORG
68 39 E0 00AE 300 BBS #FABSV_CIF+FOP,(R8),- ; special loop if CIF

```

```

62      0081 301      DOCIF
        0082 302
        0082 303 : dispatch to organization-dependent create code
        0082 304
        0082 305

      04 04 EF 0082 306 SETORG: EXTZV #FABSV_ORG,#FABSS_ORG,-
50 1D A8 0085 307 FABSB_ORG(R8),R0 : get organization code
23 A9 50 90 0088 308 MOVB R0,IFBSB_ORGCASE(R9) : and save for dispatching
25 6A 19 E0 008C 309 BBS #FWASV_NODE,(R10),20$ : branch if network operation
08 69 05 E1 00C0 310 BBC #DEVSV-SQD,IFBSL_PRIM_DEV(R9),10$ : branch if not magtape
34 69 13 E1 00C4 311 BBC #DEVSV-MNT,IFBSL_PRIM_DEV(R9),ERRDNR : error, if magtape not mounted
30 69 15 E0 00C8 312 BBS #DEV$V_DMT,IFBSL_PRIM_DEV(R9),ERRDNR : error, if magtape marked for
        00CC 313 10$: CASE TYPE=B_SRC=R0,-
        00CC 314 DISPLIST=<RMSSCREATE1,C2_BR,RMSSCREATE3>
        00D6 315
        00D6 316
        00D6 317 : error returns
        00D6 318
        00D6 319
        00D6 320

      00D6 321 RMSERR ORG
60 8F 93 00DB 322 BITB #FABSM_BIO!FABSM_BRO,- : org not supported
22 A9 00DE 323 IFBSB_FAC(R9)
29 13 00E0 324 BEQL EXIT : either bio or bro set?
FF1B' 31 00E2 325 BRW RM$BIO_CREATE : branch if not (error)
        00E5 326
        00E5 327 20$: BRW NTCRE : go do block i/o create
        00E8 328
        00E8 329 ERRRAT:
1E 10 00E8 330 BSB BSB ERROR : helper branch
        00EA 331 RMSERR_WORD RAT : both 'cr' and 'ftn' bits on
        00EC 332
        00EC 333 ERRRFM:
1A 10 00EC 334 BSB BSB ERROR : bad rfm field
        00EE 335 RMSERR_WORD RFM
        00FO 336
        00FO 337 ERRFSZ:
16 10 00FO 338 BSB BSB ERROR : fsz doesn't specify 'put'
        00F2 339 RMSERR_WORD FSZ
        00F4 340
        00F4 341 ERRMRS:
12 10 00F4 342 BSB BSB ERROR : mrs zero for fixed length rfm
        00F6 343 RMSERR_WORD MRS
        00F8 344
        00F8 345 ERRGBC:
0E 10 00F8 346 BSB BSB ERROR : GBC must be non-negative
        00FA 347 RMSERR_WORD GBC
        00FC 348
        00FC 349 ERRDNR:
0A 10 00FC 350 BSB BSB ERROR : device not mounted
        00FE 351 RMSERR_WORD DNR
        0100 352
        0100 353 ERRSYN: BBS #FABSV_OFP+FOP,(R8),-
68 3D E0 0100 354 CIF : wild card o.k. if ofp set
A6 0103 355
02 10 0104 356 BSB BSB ERROR : wild card seen
        0106 357 RMSERR_WORD SYN
        0108 357

```

50 9E 30 0108	358				
FEF2' 31 010B	359	ERROR: MOVZWL	$\@(\text{SP})+, \text{R0}$	; pick up in-line error code	
00000000'EF 17 010E	360	361	EXIT: BRW RMSCLSCU	; clean up and return to user	
0114	362	363	C2_BR: JMP RMSCREATE2		
0114	364	365	366:++		
0114	366	367	Process searchlists for a create-if operation.		
0114	367	368	The method is to loop thru the entire searchlist trying to access		
0114	368	369	the file. if we find one anywhere, we proceed to RM\$OPEN_CIF to finish		
0114	369	370	the normal \$OPEN. If no file is ever found, then we discard the		
0114	370	371	exhausted FWA, re-parse the filespec from scratch and proceed to do		
0114	371	372	an ACP create-if operation on the first element of the list.		
0114	372	373	373:--		
FEE9' 30 0114	374	DOCIF: BSBW RMSSETDID	; set up directory		
06 50 E9 0117	375	BLBC R0,10\$	; this element failed		
FEE3' 30 011A	376	BSBW RM\$ACCESS	; try to access the file		
15 50 E8 011D	377	BLBS R0,20\$	; found it, continue with OPEN		
FEDD' 30 0120	378	10\$: BSBW RM\$CHK_SLIST1	; get next element in list		
EE 50 E8 0123	379	BLBS R0,DOCIF	; and try it		
E2 51 E9 0126	380	BLBC R1,EXIT	; don't continue either		
FED4' 30 0129	381	BSBW RM\$DEALLOCATE_FWA	; release exhausted searchlist		
FED1' 30 012C	382	BSBW RMSPRFLNMALT	; get first element again		
D9 50 E9 012F	383	BLBC R0,EXIT	; failed on second attempt!		
FF7D 31 0132	384	BRW SETORG	; and proceed with create-if		
FEC8' 31 0135	385				
0138	386	20\$: BRW RMSOPEN_CIF	; continue with open		
0138	387				
0138	388	388:++			
0138	389	Network specific code for \$CREATE.			
0138	390	390:--			
0138	391				
0138	392	NTCRE:			
FEC5' 30 0138	393	BSBW NT\$ACCESS	; Perform network access function		
32 50 E9 013B	394	BLBC R0,60\$	; Branch on failure		
04 68 26 E1 013E	395	BBC #FAB\$V_SQ0+FOP,(R8),10\$	; Branch if SQ0 not specified		
10 69 3F E0 0142	396	SSB #IFBV\$V_SQ0,(R9)	; and save bit in IFAB		
06 68 39 E1 0146	397	10\$: BBS #IFBV\$V_NSP,(R9),3 \$	; Branch if task-to-task		
FEAF' 30 014A	398	BBC #FAB\$V_CIF+FOP,(R8),20\$	; Branch if CIF not specified		
19 50 E8 0151	399	BSBW NT\$OPEN	; Try to open file via remote FAL		
0154	400	BLBS R0,50\$	; Branch on successful open		
0154	401		; Note that we could check for errors		
0154	402		such as RMSS_DNF, DNR, PRV, WLK, etc.		
0154	403		to avoid getting same error on create		
FEA9' 30 0154	404	20\$: BSBW NT\$CREATE	Create file via remote FAL		
10 50 E9 0157	405	BLBC R0,40\$	Branch on failure		
01 BB 015A	406	30\$: PUSHR #^M<R0>	Save possible alternate success code		
FEA1' 30 015C	407	BSBW RMSFILLNAM	Fill in NAM block with resultant name		
01 BA 015F	408	POPR #^M<R0>	Restore status code		
05 68 39 E1 0161	409	BBC #FAB\$V_CIF+FOP,(R8),40\$	Branch if CIF not specified		
0165	410	RMSERR CREATED	Return alternate success code		
FE93' 31 016A	411	40\$: BRW RMSCREATEEXIT	Join common create code		
FE90' 31 016D	412	50\$: BRW RMSOPEN_CIF	Join common open code		
0170	413		(which will call RMSFILLNAM)		
0170	414	60\$: RMSERR CRE,R1	Establish default error code		

FE88: 30 0175 415 BSBW RMSMAPERR  
FE85: 31 0178 416 BRW RMSCREATEEXIT ; Try to map SS code to RMS code  
0178 417 ; join common create code  
0178 418 .END

\$\$PSECT_EP	= 00000000	IFBSB_RAT	= 00000051
\$\$RMSTEST	= 0000001A	IFBSB_RFMRG	= 00000050
\$\$RMS_PBUGCHK	= 00000010	IFBSL_PRIM_DEV	= 00000000
\$\$RMS_TBUGCHK	= 00000008	IFBSS_RFMR	= 00000004
\$\$RMS_UMODE	= 00000004	IFBSV_BIO	= 00000005
C2_BR	0000010E R 01	IFBSV_CREATE	= 00000032
CIF	000000AA R 01	IFBSV_NSP	= 0000003F
DEV\$V_DMT	= 00000015	IFBSV_RESTART	= 0000003B
DEV\$V_MNT	= 00000013	IFBSV_RFMR	= 00000000
DEV\$V_SQD	= 00000005	IFBSV_SQO	= 0000002D
DOCIF	00000114 R 01	IFBSW_GBC	= 00000064
ERRDNR	000000FC R 01	IFBSW_LRL	= 00000052
ERRFSZ	000000F0 R 01	IFBSW_MRS	= 00000060
ERRGBC	000000F8 R 01	NT\$ACCESS	***** X 01
ERRMRS	000000F4 R 01	NT\$CREATE	***** X 01
ERROR	00000108 R 01	NT\$OPEN	***** X 01
ERRRAT	000000E8 R 01	NTCRE	00000138 R 01
ERRRFM	000000EC R 01	PIO\$A_TRACE	***** X 01
ERRSYN	00000100 R 01	RM\$ACCESS	***** X 01
EXIT	0000010B R 01	RM\$BIO_CREATE	***** X 01
FAB\$B_FAC	= 00000016	RM\$CHK_SLIST1	***** X 01
FAB\$B_FSZ	= 0000003F	RM\$CLSCU	***** X 01
FAB\$B_ORG	= 0000001D	RM\$CREATE1	***** X 01
FAB\$B_RAT	= 0000001E	RM\$CREATE2	***** X 01
FAB\$B_RFMR	= 0000001F	RM\$CREATE3	***** X 01
FAB\$C_FIX	= 00000001	RM\$CREATEEXIT	***** X 01
FAB\$C_MAXRFM	= 00000006	RM\$DEALLOCATE_FWA	***** X 01
FAB\$C_STM	= 00000004	RM\$FILLNAM	***** X 01
FAB\$C_STMCR	= 00000006	RM\$FSETI	***** X 01
FAB\$C_STMLF	= 00000005	RM\$MAPERR	***** X 01
FAB\$C_VFC	= 00000003	RM\$OPEN_ALT	***** X 01
FAB\$L_FOP	= 00000004	RM\$OPEN_CIF	***** X 01
FAB\$M_BIO	= 00000020	RM\$PRFLNMALT	***** X 01
FAB\$M_BRO	= 00000040	RM\$SETDID	***** X 01
FAB\$M_CR	= 00000002	RM\$SCREATE	= FFFFFFFE RG 01
FAB\$M_FTN	= 00000001	RM\$S_CRE	= 0001C00A
FAB\$M_PRN	= 00000004	RM\$S_CREATED	= 00010619
FAB\$M_PUT	= 00000001	RM\$S_DNR	= 00018272
FAB\$S_ORG	= 00000004	RM\$S_FSZ	= 00018534
FAB\$V_BIO	= 00000005	RM\$S_GBC	= 000187CC
FAB\$V_BRO	= 00000006	RM\$S_MRS	= 000185D4
FAB\$V_CIF	= 00000019	RM\$S_ORG	= 0001860C
FAB\$V_CR	= 00000001	RM\$S_RAT	= 0001864C
FAB\$V_FTN	= 00000000	RM\$S_RFMR	= 00018664
FAB\$V_OFP	= 0000001D	RM\$S_SYN	= 000186D4
FAB\$V_ORG	= 00000004	SETORG	000000B2 R 01
FAB\$V_PRN	= 00000002	TPTSL_CREATE	***** X 01
FAB\$V_SQO	= 00000006		
FAB\$W_GBC	= 00000048		
FAB\$W_MRS	= 00000036		
FOP	= 00000020		
FWASV_NODE	= 00000019		
FWASV_SLPRESENT	= 00000038		
FWASV_WILDCARD	= 00000018		
IFBSB_FAC	= 00000022		
IFBSB_FSZ	= 0000005F		
IFBSB_ORGCASE	= 00000023		

```
+-----+
! Psect synopsis !
+-----+
```

## PSECT name

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
RMSRMS	0000017B ( 379.)	01 ( 1.)	PIC USR CON REL GBL NOSHR EXE RD NOWRT NOVEC BYTE
SABSS	00000000 ( 0.)	02 ( 2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

```
+-----+
! Performance indicators !
+-----+
```

## Phase

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.10	00:00:00.54
Command processing	110	00:00:00.64	00:00:06.15
Pass 1	326	00:00:10.79	00:00:30.20
Symbol table sort	0	00:00:01.54	00:00:02.41
Pass 2	87	00:00:02.09	00:00:04.44
Symbol table output	13	00:00:00.10	00:00:00.17
Psect synopsis output	1	00:00:00.05	00:00:00.25
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	570	00:00:15.31	00:00:44.16

The working set limit was 1500 pages.

60563 bytes (119 pages) of virtual memory were used to buffer the intermediate code.

There were 60 pages of symbol table space allocated to hold 1186 non-local and 23 local symbols.

418 source lines were read in Pass 1, producing 14 object records in Pass 2.

23 pages of virtual memory were used to define 22 macros.

```
+-----+
! Macro library statistics !
+-----+
```

## Macro library name

Macro library name	Macros defined
\$255\$DUA28:[RMS.OBJ]RMS.MLB;1	11
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	1
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	6
TOTALS (all libraries)	18

1294 GETS were required to define 18 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:RMSOCREAT/OBJ=OBJ\$:RMSOCREAT MSRC\$:RMSOCREAT/UPDATE=(ENH\$:RMSOCREAT)+EXECMLS/LIB+LIB\$:RMS/LIB

0329 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

